Representing Legacy System Interoperability by Extending KDM

Vegard Dehlen, Frédéric Madiot, Hugo Bruneliere

Representing Legacy System Interoperability by Extending KDM

Vegard Dehlen[1], Frédéric Madiot[2] and Hugo Bruneliere[3]

[1] SINTEF ICT
P.O.Box 124 Blindern,
N-0314 Oslo, Norway
vegard.dehlen@sintef.no

[2] MIA-SOFTWARE
4, rue du chateau de l'Eraudière BP 72 438
44324 Nantes cedex 3
fmadiot@mia-software.com

[3] ATLAS (INRIA & LINA) - University of Nantes
2, rue de la Houssinière BP 92208 - 44322, Nantes, France
hugo.bruneliere@univ-nantes.fr

Abstract. The complexity of software systems is continuously growing. An important part of this complexity issue concerns the interoperability between existing systems (i.e. legacy systems), where problems often occur due to heterogeneity in e.g. data, involved technologies or models. The Knowledge Discovery Metamodel (KDM) standardised by the Object Management Group (OMG) facilitates representation of existing systems, allowing them to be treated in a homogenous way at the model abstraction level. This paper defines a language suitable for modelling interoperability between these systems by extending KDM and introducing concepts that are specifically aimed at representing relevant interoperability information.

Keywords: interoperability, knowledge discovery, KDM, MDE

1 Introduction

The complexity of specifying interoperation between legacy systems is still a widely open issue. Their heterogeneity and distribution characteristics make them often difficult to monitor, analyse and understand. As a consequence, interoperability is difficult to represent and thus to handle There are currently many projects addressing this issue, such as the EU-funded Modelplex project which this work is part of [5]. The aim of this paper is to focus on the representation of the legacy system interoperability and to show how the Model-Driven Engineering (MDE) approach (and more especially metamodelling) is used to provide a solution in this particular context. The overall idea behind this is to be able to go from the usually heterogeneous world of systems to the homogeneous world of models in order to deal more easily with complexity.

This paper is organized as follows. Section 2 describes the main system interoperability characteristics. Section 3 is a precise definition our KDM extension for modelling systems interoperability. Section 4 concludes on the benefits of such a metamodel.

2 System Interoperability

IEEE defines interoperability as “the ability of two or more systems or components to exchange information and to use the information that has been exchanged” [1]. Research on enterprise and service interoperability has received considerable attention as reflected in projects such as IDEAS[1], INTEROP[2] and ATHENA[3]. Information & Communications Technologies.

Previous research has acknowledged that a system’s specification should be separated from its implementation. An *interface specification* serves both as a contract the system implementation must adhere to as a well as telling other systems how to interoperate with it. Traditionally, two main levels of interoperability have been distinguished [4]:

- **Signature level**. The signature (or *static*) level simply deals with the signatures of operations, i.e. names, parameters and return types.
- **Semantic level**. The more complex semantic (or *dynamic*) level deals with the meaning of operations, i.e. operational semantics, pre/post conditions, behavioural aspects of systems etc.

In addition to these two levels, the way complex systems interoperate generally relates to a combination of interoperation types. Some combinations conform to *patterns* that can be described independently of the technology. For instance, [2] have described 65 of these patterns related to integration of applications.

Within this paper, we are going to consider both the expression of *interoperability patterns* and the representation of the *interoperability relationships* corresponding to their application.

---

1 IDEAS - Interoperability Development for Enterprise Applications and Software, IST-2001-37368

2 INTEROP Network of Excellence - Interoperability Research for Networked Enterprise Applications and Software, FP6 508011. URL: http://interop-vlab.eu/

3 ATHENA - Advanced Technologies for Interoperability of Heterogeneous Enterprise Networks and their Application. URL: http://www.athena-ip.org/

3 The Interoperability Knowledge Discovery Metamodel (IKDM)

3.1 Identified Interoperability Concerns

As a part of a generic MDE solution to the described problem, this article proposes Interoperability Knowledge Discovery Metamodel (IKDM). The metamodel has been defined as an extension of the OMG Knowledge Discovery Metamodel (KDM), which is a MOF-compliant metamodel for "representing information related to existing software assets and their operational environment" [3]. Thus, the goal of KDM is to provide a common structure that facilitates interchange of data and models of legacy systems.

Based on the different types of interoperability described in Section 2 and communication with the industry partners in the Modelplex project, three main concerns to focus on have been identified:

1. *The interoperability relations between parts of each system*: the parts of the systems involved in the interaction and the role they play.
2. *The decomposition of these interoperability relations*: the sub-relations describing a relation at a lower level of abstraction.
3. *Definition of interoperability patterns*: relations between systems concerning different interoperability patterns.

By crossing the two first concerns, each kind of interoperability relationship between the top level components can be represented, and each relationship can be navigated in order to reach the primary dependencies. As a result of the third concern, the interactions between elements can then be described through a relation between the participants in the existing systems and the pattern they match with. This way of describing interoperability is independent of the kind of system and generic enough to be able to express many different possible kinds of interoperability.

3.2 The metamodel

To allow describing a large number of interoperability relationships between elements of an existing system, the concept of interoperability relationship patterns has been introduced. Such a pattern can be defined in IKDM by using three classes:

- **InteroperabilityPattern** defines a generic kind of interoperability relationship (File Transfer, Method Invocation, etc). An interoperability pattern is composed of roles and sub-patterns. Sub-patterns allow refining the pattern by decomposition.
- **InteroperabilityRole** defines the role that elements of an existing system can play in an interoperability pattern (sender, receiver, etc).
- **InteroperabilityProperty** defines properties that can characterize patterns and roles (filename, size, index, etc).

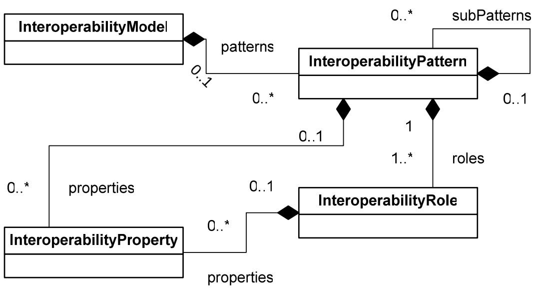

Fig. 1. IKDM Metamodel ("Pattern Expression" Part)

The interoperability patterns are used to characterize the way elements of a KDM model participate to an interoperability relationship. This kind of relationship is described with three classes:

- **InteroperabilityRelationship** defines a relationship between elements of an existing system according to a defined pattern. The aim of this class is to describe relationships at a higher level of abstraction than KDM relationships. Generally, they will be deduced from those natively described in the existing system.
- **InteroperabilityRelationshipEnd** defines the end of an interoperability relationship. It references an instance of the model element representing an aspect of the existing system (described with KDM) and an instance of *InteroperabilityRole* defining the role played by this element in the interoperability pattern. The *InteroperabilityRelationShipEnd* instances are the interface with the model of the existing systems.
- **InteroperabilityPropertyValue** defines the value for a given property set to a relationship or an end.

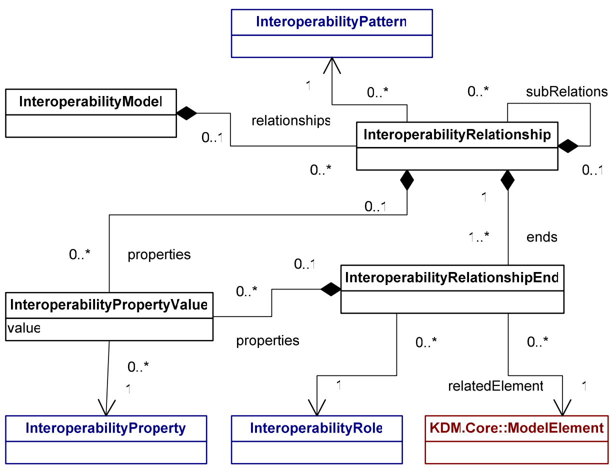

Fig. 2. IKDM Metamodel ("Interoperability Representation" Part)

IKDM is intended to support the main characteristics of interoperability, as described in Section 2. The concept of interoperability pattern (as defined within this section) has been designed to be generic enough in order to allow the description of multiple interoperability dimensions.

4 Conclusion

The work presented in this paper describes the Interoperability Knowledge Discovery Metamodel (IKDM), a KDM metamodel extension dedicated to modelling interoperability between existing systems (i.e. legacy systems). The main characteristics of system interoperability were first summarized and the IKDM metamodel, as a proposed answer to them, was then detailed.

The main contribution brought by this paper and its underlying work is the developed interoperability-specific KDM metamodel extension called IKDM. This metamodel, based on the KDM OMG standard, is particularly useful when modelling interoperability patterns and the results of their applications on real legacy systems, i.e. when modelling the interactions within a given complex system or between different systems.

The main benefit of IKDM is the possibility to define tools, based on or supporting this generic metamodel, that have the ability to manipulate models (i.e. IKDM models) independently of the type of interoperability they represent. In addition, it will be possible to identify the main interoperability patterns and express them as models in order to provide reusable libraries of such patterns.

The work will be continued during the coming months of the second phase of the Modelplex project, mainly by implementing a concrete scenario involving semi-automatic discovery of IKDM models from legacy system.

Acknowledgments. The work presented in this paper has been done in the context of the EU FP 6 project Modelplex, funded by the European Commission.

References

1. IEEE Standard Computer Dictionary: A Compilation of IEEE Standard Computer Glossaries. Institute of Electrical and Electronics Engineers (1990)
2. G. Hohpe and B. Woolf: Enterprise Integration Patterns. Addison-Wesley (2004)
3. KDM Final Adopted Specification ptc/06-06-07
4. A. Vallecillo, J. Hernández and J. M. Troya: Component interoperability. Tech. Rep. ITI-2000-37, Departmento de Lenguajes y Ciencias de la Computacion, University of Malaga (2000)
5. Modelplex – MODELling solution for comPLEX software systems, IST 34081. Official website: www.modelplex-ist.org